

# Pengenalan Gambar Menggunakan Sebagian Data Gambar

Ken Ditha Tania<sup>\*</sup>, *Fakultas Ilmu Komputer, Universitas Sriwijaya*

**Abstract**—Pengenalan gambar dengan menggunakan sebagian data gambar *query*, sebagian data gambar *query* ini bisa terjadi karena bentuk gambar *query* yang tidak utuh atau tidak sempurna gambar asli. Gambar asli ini adalah gambar yang ada didalam database Gambar *query* yang tidak utuh mungkin karena objek lain yang menutupi, atau pengambilan gambar yang tidak sempurna, atau keadaan objek itu sendiri yang mengalami perubahan. Untuk melakukan pengenalan gambar dengan kondisi seperti tersebut diatas digunakan metode ekstraksi fitur SURF.

**Kata Kunci**—Pengenalan gambar, SURF

## I. PENDAHULUAN

Pengenalan gambar pada database gambar saat ini sudah banyak sekali digunakan dalam beberapa bidang hiburan, kesehatan, perdagangan, pendidikan dan hukum. Dari waktu ke waktu penggunaannya terus saja bertambah. Seiring dengan terus bertambahnya ukuran database gambar, muncul masalah dalam pengelolaan data yg banyak tersebut terutama dalam pencarian data. Dahulu metode pencarian gambar menggunakan kata kunci. Metode ini memiliki kelemahan yaitu kata kunci yang kita inputkan kadang tidak sesuai dengan gambar yang kita inginkan. Hal ini disebabkan karena nama dari sebuah file gambar tidak dapat merepresentasikan isi dari gambar tersebut. Untuk menghindari kelemahan inilah dikembangkan metode baru yaitu CBIR (*Content Based Image Retrieval*) [1].

CBIR merupakan teknologi pencarian gambar dengan membandingkan gambar yang diinputkan sebagai *query* dengan yang ada pada database gambar (*Query By example*). CBIR dilakukan dengan membandingkan nilai jarak gambar *query* dengan gambar yang ada dalam database. Pengukuran nilai jarak gambar dilakukan berdasarkan fitur yang digunakan [1].

Pada penelitian ini akan dikembangkan sistem pengenalan gambar yang mirip dengan CBIR dengan *query* gambar yang tidak utuh atau tidak sempurna gambar asli. Gambar asli ini adalah gambar yang ada didalam database Gambar *query* yang tidak utuh mungkin karena objek lain yang menutupi, atau pengambilan gambar yang tidak sempurna, atau keadaan objek itu sendiri yang mengalami perubahan. Untuk melakukan pengenalan gambar dengan kondisi seperti tersebut diatas digunakan metode ekstraksi fitur SURF. SURF saat ini sering digunakan pada pengenalan objek [2],

[3]. Dengan fitur ini diharapkan akan dapat dilakukan proses ekstraksi ciri dengan hasil yang baik pada saat proses pencocokan.

Tujuan dari penelitian ini adalah untuk mengetahui efektivitas penggunaan fitur SURF pada suatu pengenalan gambar dengan *query* gambar yang tidak utuh.

## II. TINJAUAN PUSTAKA

Suatu gambar memiliki ciri yang berbeda satu dan yang lainnya bergantung pada karakteristik yang menonjol pada gambar tersebut. Sebagai contoh, bunga matahari dan bunga melati dibedakan melalui perbedaan warnanya, kain dan kertas dapat dibedakan dari teksturnya, sedangkan gambar lingkaran dengan gambar kotak dibedakan melalui bentuknya. Masing-masing ciri dasar dari gambar ini didapatkan melalui proses ekstraksi ciri yang tidak mudah, karena satu gambar dapat mempunyai *multiple feature*. Proses ekstraksi ciri yang baik menentukan keberhasilan dalam membangun aplikasi pengenalan image

Pencarian suatu gambar dengan membandingkan gambar *query* dengan gambar yang ada didatabase secara umum dibangun dengan melihat karakteristik dari suatu gambar atau dengan kata lain dengan melihat ciri dari gambar tersebut.

Ciri merupakan suatu tanda yang khas, yang membedakan antara satu gambar dengan gambar yang lain. Pada tulisan ini ciri yang di pakai adalah fitur SURF.

### A. Fitur SURF

Fitur SURF ini merupakan *keypoint* dari sebuah gambar. *Keypoint* adalah titik-titik dari sebuah gambar yang nilainya tetap ketika mengalami perubahan skala, rotasi, blurring, pencahayaan, dan juga perubahan bentuk. Perubahan bentuk ini bisa terjadi karena bentuk gambar *query* yang tidak utuh atau tidak sempurna gambar yang ada didalam database gambar tersebut. Gambar *query* yang tidak utuh mungkin karena objek lain yang menutupi, atau pengambilan gambar yang tidak sempurna, atau keadaan objek itu sendiri yang mengalami perubahan. Agar invarian terhadap perubahan skala maka proses yang dilakukan pertama kali adalah membuat ruang skala (*scale space*).

#### Ruang Skala (*Scale Space*)

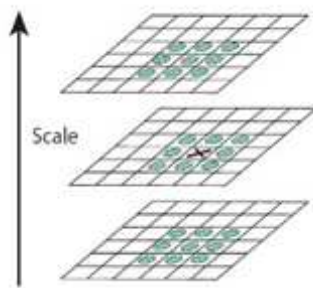
*Scale space* terbagi kedalam bilangan yang disebut *octave*. Setiap *octave* merepresentasikan respon filter yang diperoleh dengan melakukan proses konvolusi gambar yang diinputkan dengan ukuran filter yang menaik.

Ken Ditha Tania adalah staf pengajar dan peneliti di Jurusan Sistem Informasi, Fakultas Ilmu Komputer, Universitas Sriwijaya, Inderalaya, Ogan Ilir, Sumatera Selatan (e-mail: ken.tania@yahoo.com).

### Lokalisasi Keypoint

Lokalisasi *keypoint* dilakukan dengan beberapa proses. Proses pertama, menentukan *threshold* untuk *keypoint*. Ketika *threshold* dinaikan, jumlah *keypoint* yang terdeteksi lebih kecil begitupun sebaliknya. Oleh karena itu, *threshold* bisa disesuaikan pada setiap aplikasi.

Proses berikutnya *non-maxima suppression*, proses ini dilakukan untuk mencari sekumpulan calon *keypoint* dengan membandingkan tiap-tiap *pixel* gambar pada *scale space* dengan 26 tetangga. 26 tetangga *pixel* itu terdiri atas 8 titik di *scale* asli dan 9 titik di tiap-tiap *scale* diatas dan dibawahnya. Proses inilah yang menghasilkan *keypoint* dari suatu gambar. Gambar 1 menunjukkan *non-maxima suppression*.



Gambar 1. *Non-maxima suppression* [3]

Proses terakhir yaitu proses mencari lokasi *keypoint* menggunakan interpolasi data yang dekat dengan *keypoint* hasil proses sebelumnya. Ini dilakukan dengan mencocokkan *quadratic* 3D yang diajukan oleh Brown [4].  $H(x, y, \sigma)$  adalah determinan *Hessian*, didefinisikan sebagai berikut [3]:

$$H(x) = H + \frac{\partial H^T}{\partial x} x + \frac{1}{2} x^T \frac{\partial^2 H}{\partial x^2} x \quad (1)$$

Lokasi ekstrim yang diinterpolasi,  $\hat{x} = (x, y, \sigma)$ , ditemukan dengan mencari turunan dari fungsi diatas dan diberi nilai nol, sehingga:

$$\hat{x} = -\frac{\partial^2 H}{\partial x^2} \frac{\partial H}{\partial x} \quad (2)$$

### Deskriptor Keypoint dan Proses Matching

Deskriptor merupakan daerah piksel disekitar *keypoint* yang dihasilkan. Deskriptor menggambarkan distribusi intensitas piksel tetangga di sekitar *keypoint*.

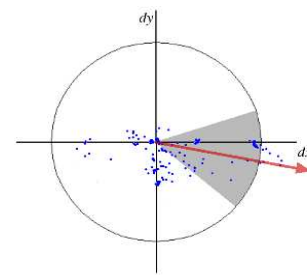
Proses pertama yang dilakukan adalah mencocokkan orientasi yang dihasilkan berdasarkan informasi dari daerah yang berbentuk lingkaran disekitar piksel yang menjadi *keypoint*.

Kemudian proses berikutnya membuat daerah berbentuk kotak pada orientasi yang terpilih dan mengekstrak deskriptor SURF dari daerah tersebut. kemudian proses *matching* fitur antara dua gambar dilakukan. Berikut ini dua proses perhitungan deskriptor *keypoint* yang akan dijelaskan lebih detail.

### Pembuatan Orientasi

Pembuatan orientasi dilakukan dengan menghitung respon Haar wavelet dalam arah x dan arah y pada daerah piksel tetangga disekitar *keypoint* yang berbentuk lingkaran dengan jarak  $6s$ , dimana  $s$  adalah *scale* dari *keypoint* yang dideteksi. Ukuran wavelet ditentukan menjadi  $4s$ . Filtering dipercepat prosesnya menggunakan integral gambar, dan keluaran yang dihasilkan direpresentasikan sebagai titik-titik pada ruang dengan respon horizontal sepanjang sumbu *axis* dan juga respon vertikal sepanjang sumbu koordinat.

Orientasi dominan dihitung menggunakan jumlah semua respon dalam pergeseran orientasi dengan ukuran window  $\pi/3$  (seperti ditunjukkan pada Gambar 2), kemudian respon horizontal dan vertikal dalam *window* tersebut dijumlahkan. Dari kedua respon yang dijumlahkan sehingga menghasilkan sebuah vektor orientasi lokal. Vektor terpanjang inilah yang merupakan orientasi *keypoint*.



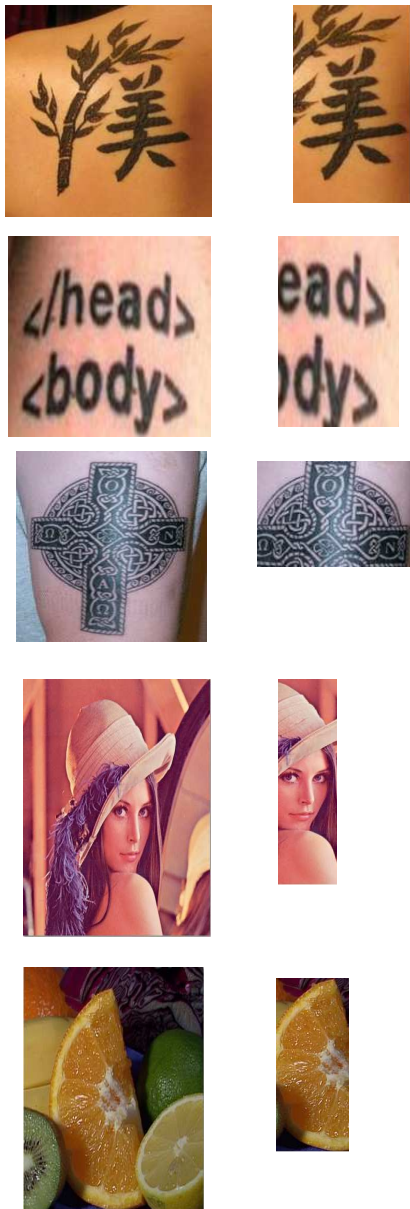
Gambar 2. Jendela pergeseran orientasi [3]

### Deskriptor Berdasarkan Jumlah Respon Haar Wavelet

Proses ekstraksi deskriptor dilakukan dengan membuat daerah kotak persegi disekitar *keypoint*, dimana *keypoint* sebagai pusat dari daerah kotak persegi tersebut, dan orientasinya disekitar orientasi yang ditentukan dan ditunjukkan pada Gambar 2. Ukuran *window* yang diambil  $20s$ .

Kemudian, respon wavelet  $dx$  dan  $dy$  dijumlahkan untuk setiap sub-region. Hal ini akan memberikan informasi tentang polar dari perubahan intensitas, dan juga akan dihasilkan jumlah nilai absolut dari respon  $|dx|$  dan  $|dy|$ . Masing-masing sub-region mempunyai 4 dimensi deskriptor vektor  $v$ , yaitu  $dx, dy, |dx|$  dan  $|dy|$ . Untuk  $4 \times 4$  sub-region, maka panjang vektor deskriptornya berjumlah 64 [3].

$$v = \left( \sum d_x, \sum d_y, \sum |d_x|, \sum |d_y| \right) \quad (3)$$



Gambar 3. Contoh hasil gambar eksperimen. Sebelah kanan data *query* dan sebelah kiri data gambar didalam *database*

#### B. Modul Matching Fitur SURF Menggunakan Teknik FLANN

Metode Fast Library Approximated Nearest Neighbor (FLANN) digunakan untuk *matching* fitur SURF. Fitur SURF ini terdiri dari *keypoint* dan *descriptor* yang berupa vektor. Untuk satu gambar di basis data terdapat banyak *cluster* untuk masing-masing fitur SURF. *Cluster* ini dibuat otomatis dengan KNN menggunakan tipe indeks kd tree, dimana KNN ini akan mencari jarak terkecil antara vektor fitur dengan vektor pada *cluster*.

Proses pencocokan fitur pada gambar *query* dan fitur pada gambar dalam basis data, vektor *keypoint* dan *descriptor* pada gambar *query* akan dicocokkan nilainya menggunakan KNN search. KNN search akan mencari *cluster* pada basis data yang nilai vektor deskriptornya paling dekat jaraknya dengan vektor *descriptor* pada gambar *query*. Setelah

*cluster* didapat, kemudian akan dicari nilai vektor *descriptor* pada *cluster* tersebut yang sama atau paling dekat dengan vektor *descriptor* pada gambar *query*. Jika ada yang sama maka ada satu *keypoint* yang cocok antara kedua gambar tersebut. Kemudian, untuk setiap gambar pada basis data, dihitung jumlah *keypoint* yang paling banyak cocok. Jumlah *keypoint* yang paling banyak cocok ini merupakan gambar pada basis data yang paling mirip dengan gambar *query*.

### III. HASIL DAN PEMBAHASAN

Eksperimen dilakukan dengan 30 gambar. Skenario eksperimen dilakukan dengan 30 gambar asli dijadikan *database* dan 30 gambar di *database* dipotong sebagian untuk dijadikan data *test* atau data *query*. Pemotongan gambar untuk dijadikan *query* dilakukan dengan menggunakan perangkat lunak Paint. Pemotongan dilakukan secara sembarang.

Hasil yang didapat 27 gambar dapat dikenali dengan benar, sedangkan 3 gambar yang lainnya tidak dapat dikenali. Gambar 3 menunjukkan contoh gambar hasil eksperimen

### IV. KESIMPULAN

Dari percobaan yang dilakukan dapat diambil kesimpulan sebagai berikut, pengenalan gambar menggunakan sebagian data gambar sebagai data *query* gambar dengan metode SURF menghasilkan tingkat pengenalan 90%.

Pengenalan gambar menggunakan sebagian data gambar sebagai data *query* gambar dengan metode SURF tidak berhasil mengenali dikarenakan objek dalam gambar tersebut terlalu kecil.

### REFERENSI

- [1] Rahadiani, L., Ruli Manurung, and Aniati Murni. Clustering Batik Images based on Log-Gabor and Colour Histogram Features. [Online]. Available: <http://www.cs.ui.ac.id/files/icacsis2009/pdf/17.pdf>
- [2] M. Brown and D.G. Lowe. Invariant features from interest point groups. *British Machine Vision Conference*, Cardiff, Wales, pages 656–665, 2002
- [3] Marius Muja and David G. Lowe. Fast approximate nearest neighbors with automatic algorithm configuration. In *International Conference on Computer Vision Theory and Applications (VISSAPP'09)*, pages 331–340, 2009
- [4] H. Bay, B. Fasel, and L. van Gool. Interactive museum guide: Fast and robust recognition of museum objects. In *Proceedings of the first international workshop on mobile vision*, May 2006
- [5] A. Neubeck and L. Van Gool. Eficient non-maximum suppression. In *ICPR*, 2006
- [6] H. Bay, T. Tuytelaars, and L. Van Gool. SURF: Speeded up robust features. In *ECCV*, 2006
- [7] A. Witkin. Scale-space Itering, *Int. Joint Conf. Artificial Intelligence*, 2:1019{1021, 1983.
- [8] H. Bay, T. Tuytelaars, and L. Van Gool. Speeded-Up Robust Features (SURF), *Computer Vision and Image Understanding (CVIU)*, Vol. 110, No. 3, pp. 346--359, 2008
- [9] Notes on the OpenSURF Library [online]. Available: <http://opensurf1.googlecode.com/files/OpenSURF.pdf>
- [10] M. Brown and D.G. Lowe. Invariant features from interest point groups. *British Machine Vision Conference*, Cardiff, Wales, pages 656–665, 2002